20807070 20070860

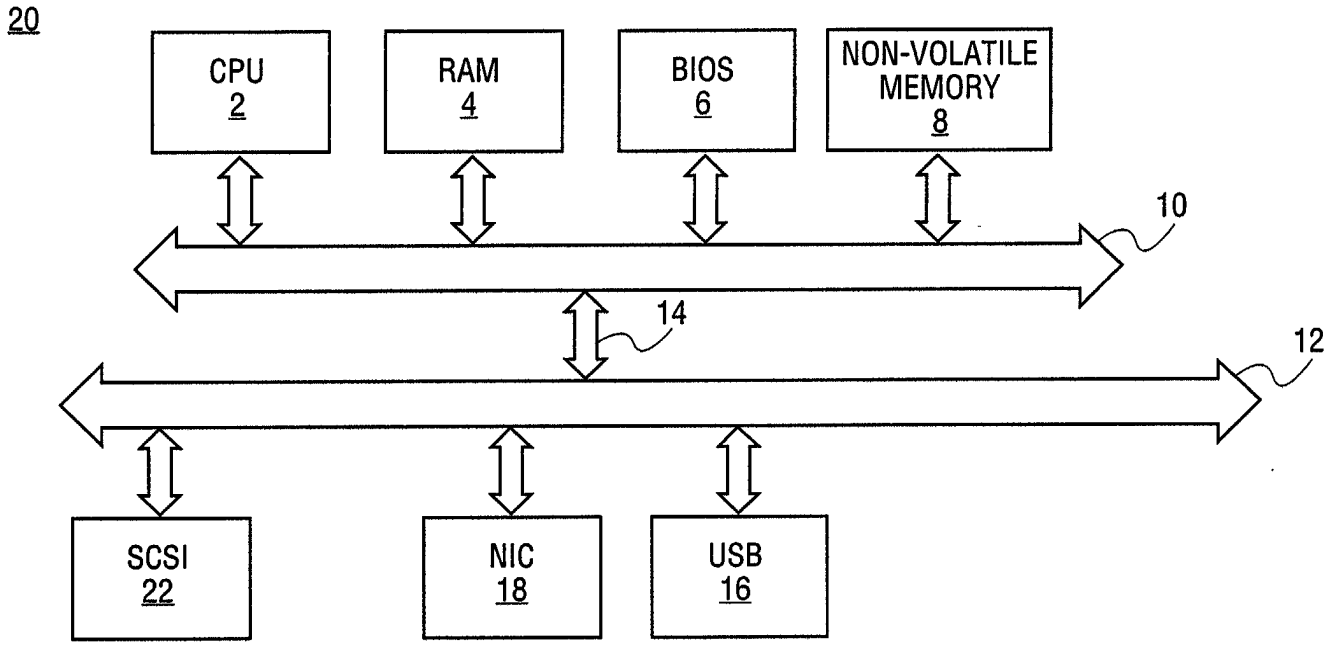


FIG. 1

20807O" 2007838607

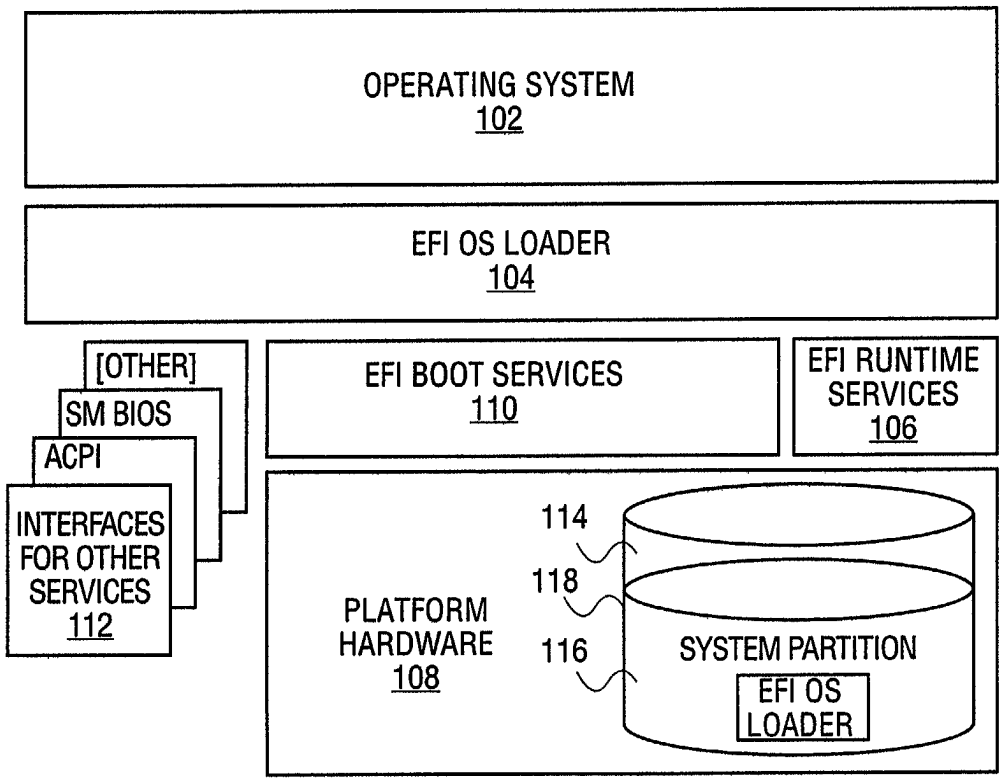


FIG. 2

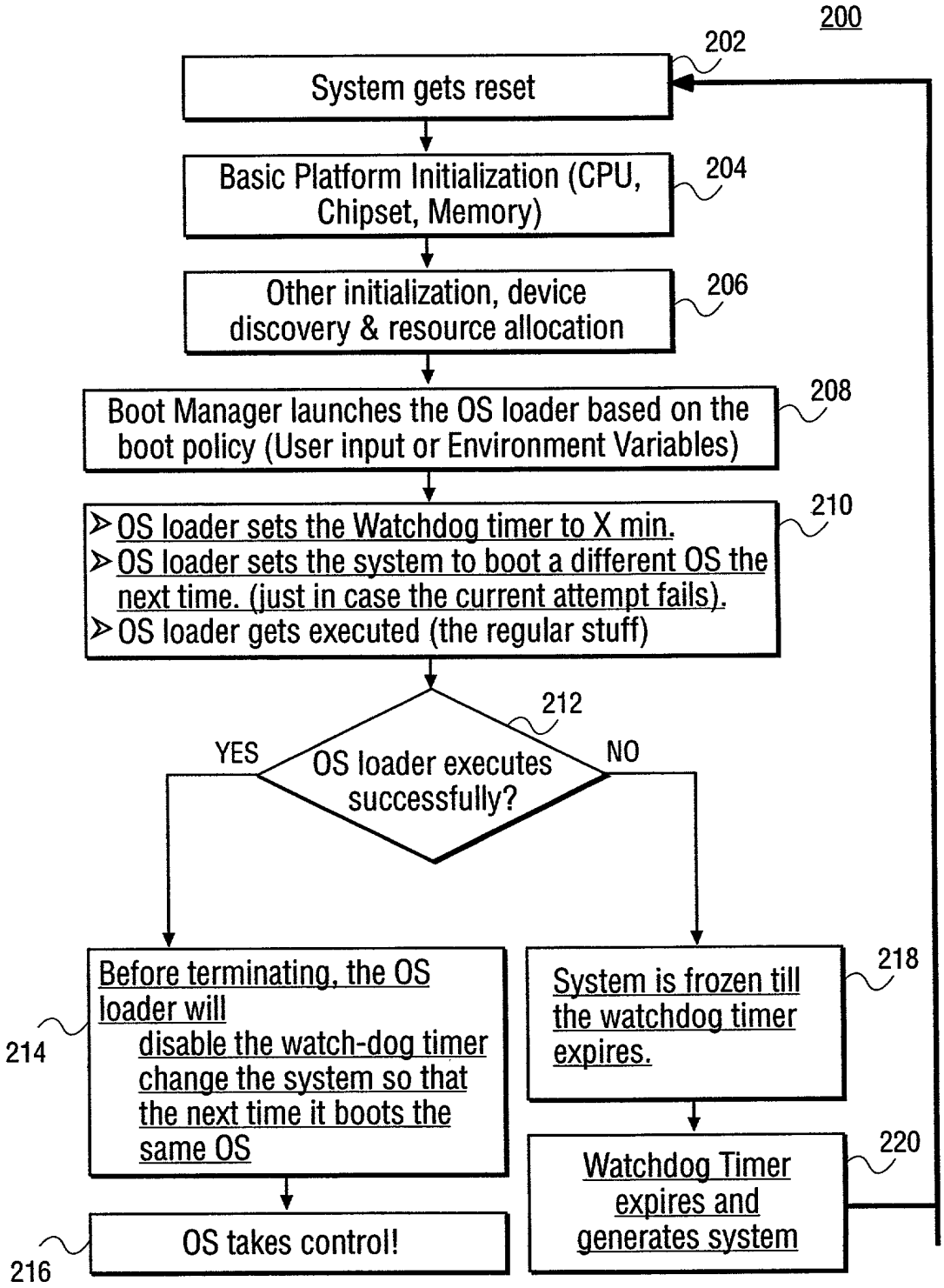

200
202
System gets reset
Basic Platform Initialization (CPU, Chipset, Memory)
204
Other initialization, device discovery & resource allocation
206
Boot Manager launches the OS loader based on the boot policy (User input or Environment Variables)
208
OS loader sets the Watchdog timer to X min.
OS loader sets the system to boot a different OS the next time. (just in case the current attempt fails).
OS loader gets executed (the regular stuff)
210
212
OS loader executes successfully?
YES
NO
Before terminating, the OS loader will
disable the watch-dog timer
change the system so that the next time it boots the same OS
214
OS takes control!
216
System is frozen till the watchdog timer expires.
218
Watchdog Timer expires and generates system
220

FIG. 3

300

```
// Each OS loader can choose their own timer duration dependent on how long they need to boot
SetWatchDogTimer(360,    //set the watchdog timer for 6min - 5 * 60 seconds
                 0xffffff, //OEMs can choose any number other than those between 0x0000-0xFFFF, 0);
```

302

```
// Set the BootNext variable to the appropriate value. It will be used in the next reboot if the current attempt fails.
// BootCurrent will have the OS loader that will be used for the current boot. This will set by the boot manager.
// So get that value in the variable "CurrentlySelectedBootLoader):
GetVariable (L"BootCurrent".....,CurrentlySelectedBootLoader);
// We have to find out what OS loader is next to the currently selected one in the system's BootOrder.
GetVariable (L"BootOrder".....,SystemBootOrder);
// Get the number of elements in SystemBootOrder. BootOrder is an array of UINT16s.
int LengthOfArray = sizeof (SystemBootOrder) / sizeof (UINT16);
// Now scan through the list of boot options in Boot Order. (BootOrder is an ordered list of boot options)
for (int i=0; i < LengthOfArray; i++ )
{
        if (i == LengthOfArray - 1) //Extreme condition when no other boot options are available to try.
        {       Exit(); //Exits to the firmware shell it has exhausted all the possible OS boot options.
                // This behavior can be customized by the OEMs to alert an administrator or do other things.
        }
        // if the given element in the array is the currently selected OS loader stored in BootCurrent, then...
        if (SystemBootOrder[i] == CurrentlySelectedBootLoader)
        {       // if the current boot attempt fails, we must boot the next OS as specified in the BootOrder
                // So, set the BootNext variable to the appropriate value i.e., the next element in the BootOrder
                // list
                SetVariable(L"BootNext",.....sizeof(UINT16), SystemBootOrder[i+1]);
                break;
        }
}
```

304

TO FIG. 4B

FIG. 4A

20030703 200T3860

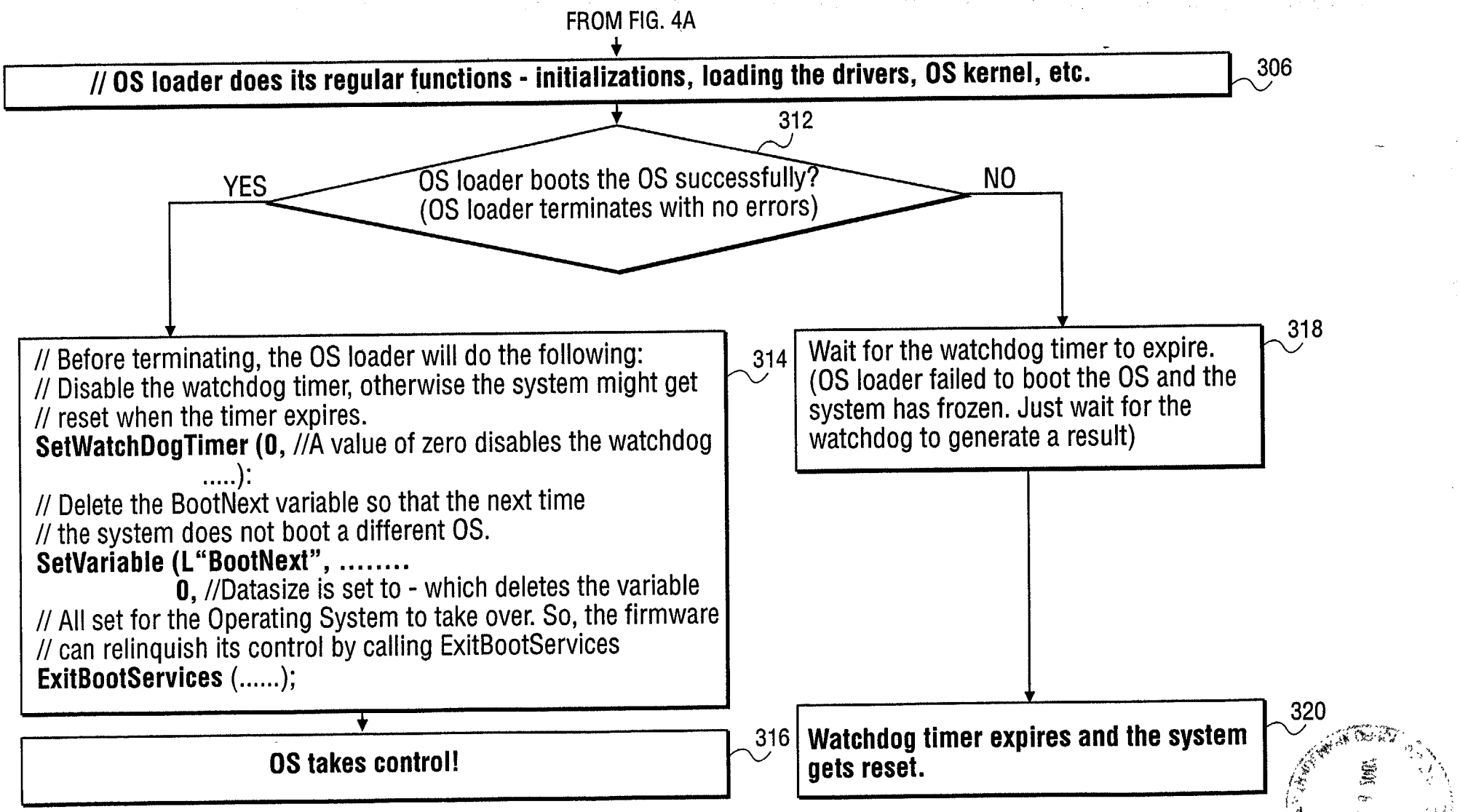


FIG. 4B